

# Scale-Invariant Line Descriptors for Wide Baseline Matching

Bart Verhagen

`bart.verhagen@tass.be`

TASS, Belgium

VISICS, KU Leuven, Belgium

Radu Timofte

`timofte@vision.ee.ethz.ch`

VISICS, KU Leuven, Belgium

CVL, ETH Zürich, Switzerland

Luc Van Gool

`vangool@esat.kuleuven.be`

VISICS, KU Leuven, Belgium

CVL, ETH Zürich, Switzerland

## Abstract

*In this paper we propose a method to add scale-invariance to line descriptors for wide baseline matching purposes. While finding point correspondences among different views is a well-studied problem, there still remain difficult cases where it performs poorly, such as texture-less scenes, ambiguities and extreme transformations. For these cases using line segment correspondences is a valuable addition for finding sufficient matches. Our general method for adding scale-invariance to line segment descriptors consist of 5 basic rules. We apply these rules to enhance both the line descriptor described by Bay et al. [1] and the mean-standard deviation line descriptor (MSLD) proposed by Wang et al. [14]. Moreover, we examine the effect of the line descriptors when combined with the topological filtering method proposed by Bay et al. and the recent proposed graph matching strategy from K-VLD [6]. We validate the method using standard point correspondence benchmarks and more challenging new ones. Adding scale-invariance increases the accuracy when confronted with big scale changes and increases the number of inliers in the general case, both resulting in smaller calibration errors by means of RANSAC-like techniques and epipolar estimations.*

## 1. Introduction

Stereo wide baseline matching is a problem within computer vision in which correspondences between two views are found in a systematic way. These two views can differ in viewpoint, rotation, scale, illumination and camera parameters. The fundamental matrix can be derived for the two views given at least 7 point correspondences and the epipolar geometry with at least 5 point correspondences [5].

Wide baseline matching has been largely solved in the case of textured scenes. For instance, MSER [8] and Lowe's scale-invariant feature transform (SIFT) algorithm [7] are robust methods to efficiently find matches between such images. Several "SIFT-like" descriptors have been docu-

mented. We call "SIFT-like" the methods that either use the SIFT descriptor or propose a descriptor using the same principles as SIFT. In combination with graph matching methods or topological filters [3], often complemented with RANSAC-based [4] methods, SIFT and its derivatives perform well when applied on two or more views, even when additional information like epipolar geometry is missing. Graph matching methods take the geometric consistency of the correspondences into account. As a result they tend to have a high complexity [2, 6], especially the higher order graph matching methods. Liu and Marlet [6] describe a graph matching method in which only consistency with the local neighborhood is checked, resulting in a much faster algorithm. Moreover, most graph matching methods require a high inlier rate in order to work correctly.

Topological filters also consider the layout of the correspondences, but enforce weaker constraints. As a result, the execution time of topological filtering is significantly smaller, but the matching performance decreases as well. Topological filters have been shown to resist a higher percentage of outliers than most RANSAC-based algorithms do.

On the other hand, wide baseline stereo matching for texture-less scenes still represents a much harder problem: interest point detectors will not provide enough correct correspondences in order to be able to estimate the epipolar geometry correctly. Such scenes typically occur indoors and therefore are often characterized by many straight lines. It is in these cases that wide baseline matching using line segments, often in combination with interest point methods, are particularly useful. However, using line segments introduces some problems of its own, the most prominent being the fact that the endpoints of a line segment are inaccurately localized. Therefore more statistical measures need to be taken to describe line segments than is the case for point or region descriptors.

Bay *et al.* [1] propose a method that describes line segments using color histograms and a topological filter. While yielding distinctive information, the drawback of using color information for matching is that the method may

not be sufficiently robust to illumination transformations. Moreover, Bay *et al.*'s color histogram descriptor does not take into account the scale of the objects in the view.

Wang *et al.* [14], on the other hand, use a SIFT-like strategy for describing line segments called the mean-standard deviation line descriptor (MSLD). Similar to SIFT, MSLD is more robust than Bay *et al.*'s descriptor to changes in illumination conditions. MSLD does not take scale into account either. Zhang and Koch [16] extended the MSLD descriptor by adding geometric constraints and a topological filter to eliminate wrong matches. While the geometric constraints applied after the MSLD algorithm are scale-invariant, the descriptor itself is not. Wang *et al.* [15] extended the MSLD descriptor to also make use of color information, thereby slightly increasing the matching accuracy for baseline images with similar illumination conditions.

In this paper we propose an approach to add scale-invariance to line segment descriptors for wide baseline matching. We show that the scale-invariant descriptors cope better with larger scale changes, while maintaining its robustness under other image transformations w.r.t. their scale-variant counterparts without scale spaces. In general these scale-invariant descriptors will yield more inliers. Despite many of these inliers being redundant, this allows better epipolar estimations. The introduced redundancy also yields more efficient and effective topological filtering and graph matching, which in some methods, *e.g.* our slightly modified implementation of Bay *et al.*'s line segment matching algorithm, results in a speed up of the algorithm.

## 2. The scale-invariant line segment descriptor

### 2.1. Scale-invariant line segment matching: the principles

Wide baseline stereo matching often has to deal with scale changes. It is therefore important to take scale into account when matching. MSER and SIFT adapt to those scale changes, but line descriptors still largely lack such capacity. Scale-invariant wide baseline methods typically use a scale space for this purpose [7]. Unfortunately building a scale space requires  $N$  line segment detections and  $(N - 1)$  image rescales, where  $N$  is the number of scale levels in the image pyramid. This tends to slow down and increase the complexity of the algorithm w.r.t. the scale-variant case.

Therefore we introduce a faster but less robust alternative to extract the scale by using the line segments detected on the lowest scale only. A pair of line segments can be the edges of a thicker line. More details in Section 2.2.

The main principle when matching line segments is that corresponding descriptors must describe corresponding regions w.r.t. the line segments. In order to find these regions we derive five rules for extending a line descriptor to be-

come scale-invariant:

1. Preserve the principles of the original descriptor and apply them whenever possible.
2. Real world parallel lines do not necessarily stay parallel under baseline changes. Most lines parallel in the real world become lines that intersect into one point and we use this in the new descriptor (see also rule 5).
3. Distances between points change w.r.t. scale  $s$ . Therefore, given  $d$  the distance between the line segment and the region in the original descriptor, this distance scales as  $d(s + 1)$ , such that the distance never goes to zero. This is a result of the fact that one-pixel wide lines are defined to have scale zero ( $s = 0$ ).
4. As a result of the previous rule, the width and height of a region both scale with a factor of  $s + 1$  as well. Other parameters that depend on the size of a region scale accordingly.
5. Take into account the global (assumed) scale deformation. The same deformation should be applied to the distances and the regions of the descriptor. Note that this will often lead to a similar conclusion as rule 2 since deformed parallel lines will indeed not stay parallel in most cases, but will intersect.

In section 2.3 and 2.4 these rules will be illustrated on two scale-variant line segment descriptors. In these sections it will be shown that these rules should be applied multiple times on several aspects of the descriptor. First, section 2.2 describes our method for extracting the scale in more detail.

### 2.2. Line segment scale extraction: the fast approach

Typically scale-invariance is achieved using scale spaces. In this paper we will use a different approach that will apply a single line segment detection.

In this approach, we will focus on line segment pairs that lie close to each other in the image, capable to be edges of thicker line segments (or stripes). The typical case will be (near-) parallel line segments. Therefore the angle between the two line segments ( $\theta$  in Figure 1a) is sufficient to describe the deformation. We define the *scale angle*  $\phi$  as the complementary angle of  $\frac{\theta}{2}$  (Figure 1a).

A first step in our method of extending scale-invariance to line segment detectors is to find the thickness of stripe-like structures in the images. A stripe is a line segment with a width of multiple pixels. Due to deformation the stripe can become trapezoidal, resulting in a so-called stripe-like structure. In order to find those stripe-like structures, we first detect all line segments using a traditional line segment detector [13]. This results in one-pixel wide lines that

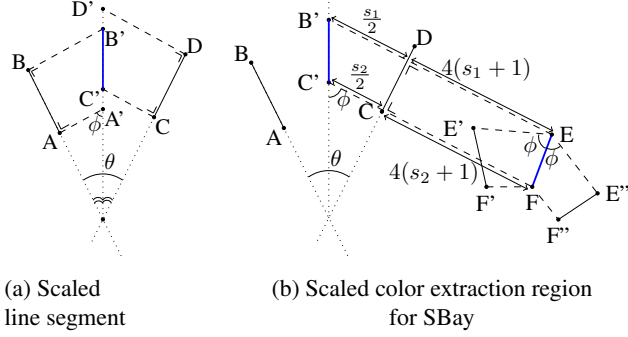


Figure 1: Illustrations on how scaled lines (a) and their corresponding scaled regions in SBay (b) are constructed.

depict the edges of the stripe-like structure rather than the stripe-like structure itself. Next we select all pairs of line segments for which  $\theta$  (see Figure 1a) as well as the mean of the orthogonal distances between the line segment endpoints and the other line segment are below selected thresholds. A further pruning of pairs will follow, but already here we are quite selective. The threshold applied to the orthogonal distance is kept rather small, i.e. 50 pixels for images of  $800 \times 600$  pixels. The threshold on  $\theta$  is taken to be 10 degrees.

Further pruning is applied by considering the overlap between the projections of the two line segments onto the bisector line, as shown in Figure 1a. In this figure the overlap is given by the line with endpoints  $B'$  and  $C'$ . We compute the scale at these endpoints by considering the sum of the orthogonal distances of the endpoint to the original line segments ( $2d(B, B')$  and  $2d(C, C')$  in this case). These scales are indicated as  $s_1$  and  $s_2$  in Figure 1b. The two endpoints, the scale in each of the endpoints and the scale angle describe the scaled line  $B'C'$ . These five parameters will play a role when describing our scale-invariant adaptations for both the Bay descriptor (Section 2.3) and the MSLD descriptor (Section 2.4).

When the scale of the images decreases, the stripe will eventually become a one-pixel wide line segment. Such one-pixel wide line segments are considered to have scale 0 for both endpoints and a scale angle of  $\frac{\pi}{2}$ .

### 2.3. Scale-invariant Bay line segment matching descriptor

Bay *et al.* [1] propose a descriptor in which a color histogram is constructed using the color information along two parallel line segments with the same length at a transversal distance of 4 pixels from the originally detected line segment. We adapt this descriptor to become scale-invariant by applying the rules mentioned in section 2.1. In this text we will further refer to the line segment algorithm proposed by Bay *et al.* as *Bay* and to our adapted algorithm as *scale-*

*invariant Bay* or *SBay*.

For the SBay descriptor, first the middle line of the region from which we want to extract the color information needs to be found (rule 1). This is done by starting in an endpoint of the scaled line segment and go for a distance of  $d(s+1) + \frac{s}{2}$  (rule 3) in the direction given by the scale angle (rule 2 and 5). This is done for each endpoint of the scaled line segment in both the  $+\phi$  and  $-\phi$  direction (rule 1, 2 and 5). This is illustrated in Figure 1b for one direction, resulting in line EF. Here  $d$  is chosen to be 4, as was suggested by Bay.

Next, we have to extract the color information around this line. Since Bay extracts a one-pixel wide line with the same length to describe an original line segment of the same width, SBay also extracts the color information within a ‘line’ that has the same ‘thickness’ and length as the scaled line. Since scaled lines are in fact regions, color extraction will also occur from an entire region that scales with the scale of the scaled line (rule 4). As we already know the deformation parameters from the scaled line generation phase, we can simply apply kind of the reverse operations on the earlier constructed line for constructing the extraction region. First, starting from the middle line EF, two flanking line segments need to be constructed. To that end a distance of  $\frac{s}{2}$  in the direction of  $+\phi$  and  $-\phi$  w.r.t. middle line EF for each endpoint needs to be taken (rule 2 and 5), resulting in points  $E'$ ,  $E''$ ,  $F'$  and  $F''$  in Figure 1b. The region from which SBay will extract the color information for the descriptor is the resulting hexagon (E-E'-F'-F''-E'') in Figure 1b).

Note that Bay applies spline interpolation on the intensities after they are extracted in order to suppress noise. This requires SBay to do a 2D-spline interpolation. However, empirical results from Bay’s algorithm seem to indicate that the spline interpolation degrades the performance of the line matching method. Therefore noise suppression on the extracted colors for both Bay and SBay are omitted in our implementation.

### 2.4. Scale-invariant MSLD

The MSLD algorithm proposed by Wang *et al.* [14] applies a SIFT-like approach to nine parallel lines where the fifth line (in the middle) is the original line segment. The lines are separated by a distance of 5 pixels. The SIFT-like method is applied to a subregion of  $5 \text{ px} \times 5 \text{ px}$  centered around each pixel on each line. The subregions are mapped on four different directions (+ and - in horizontal and vertical direction). Next the mean and standard deviation over each direction of all points on each line are taken and normalized. The MSLD descriptor does not use a scale space and is not scale-invariant. The MSLD descriptor can be adapted to become scale-invariant by once again applying the rules of Section 2.1 on several aspects of the algorithm.

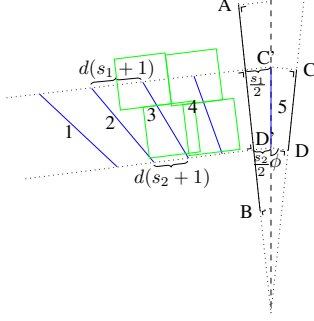


Figure 2: Subregion generation of SMSLD. The right side is the mirror of the left.

In this text we will further refer to this adapted algorithm as *scale-invariant mean-standard deviation line descriptor* or *SMSLD*. When discussing the SMSLD algorithm, the nine lines are numbered from top to bottom and from left to right (see Figure 2).

Note that, in order to describe the exact same region for different scales,  $\phi$  should be used to additionally change the direction between every line. As a result the lines would be no longer parallel but would all intersect in the same point (rule 5). However, the placement of the lines is simplified to an approach similar to how the middle line of the region from which to extract the colors in SBay (see Section 2.3) is constructed. Lines 4 and 6 are at a distance  $d(s+1) + \frac{s}{2}$  from the line segment (rule 3) in the direction denoted by  $-\phi$  and  $+\phi$  respectively (rule 2 and 5). Lines 1-2-3-4 and 6-7-8-9 are each time separated by a distance  $d(s+1)$  (rule 3), measured in the same direction as line 4 and 6 respectively.  $d$  is chosen to be 5 pixels, as proposed by Wang *et al.* [14].

To speed up the description, not every pixel on each of these lines is taken as a center for a subregion: the centers of each subregion on the same line are separated by a distance of  $s+1$ . The size of each subregion is adjusted to the scale as well: the width and height of each subregion is  $d(s+1)$  (rule 4). A further speedup of the algorithm is accomplished by keeping the orientation of each subregion the same as if it was a subregion for the originally detected line segment on the same side (see Figure 2: the orientation of the subregions for line 1 - 2 - 3 - 4 are oriented as if they are subregions of line AB). Only for the middle line (line 5) the orientation of the subregions is adapted to the line itself.

Note that SMSLD needs a big region at each side of a line segment in order to be able to describe the line segment. Line segments whose descriptor regions contain too many invalid points are rejected. To increase the number of valid scaled lines we add 50 pixels to all sides of the image and repeat the 50 closest lines or columns in reverse order.

### 3. Evaluation

The proposed descriptors are evaluated on standard benchmarks and on some typical (challenging) indoor scenes<sup>1</sup>. The original descriptors are provided as a reference as well. In addition we provide the results for K-VLD [6], a state-of-the-art region and point descriptor, and, where meaningful, SIFT [7]. The impact when combined with a topological filter [1] or the K-connected graph matching filter [6] is also examined. More details about the evaluation follows in Section 3.1.

#### 3.1. Evaluation units

The following units are combined in order to evaluate the SBay and SMSLD descriptors:

**LSD** The line segment detector version 1.5 implementation of Grompone von Gioi *et al.* [13] is used for detecting line segments. When used with Bay and SBay, the scale and EPS parameter are adjusted until the number of line segments is below 400. For MSLD and SMSLD the scale is set to 1 while all other parameters are set at their default values. In the case of MSLD and SMSLD all found line segments are kept.

**Bay** Our MATLAB implementation of the line segment matching method proposed by Bay *et al.* [1] up to the correspondence boosting. Note that this is not the entire algorithm as proposed by Bay *et al.* The remaining filtering stages are omitted since these can be applied to all line segment descriptors for improving their accuracy as well. The snapping part is also omitted. This method also includes the collinear line merging method described by Bay *et al.* The default parameters as described by Bay *et al.* are used for the algorithm, however due to efficiency considerations only the 200 best soft matches are kept before the topological filtering phase.

**SBay** Our MATLAB implementation of Bay adapted to become scale-invariant as described in Section 2.3. Scaled lines are generated using the parameters mentioned in Section 2.2.

**MSLD** The original C++ implementation of the MSLD algorithm provided by the authors [14]. No collinear line merging is used. Using an adapted version of Bay *et al.*'s collinear line merging method results in longer matched line segments, but without significant impact on the accuracy and is therefore omitted. The MSLD descriptor is set to only use straight line segments for matching, the other parameters are the default ones. No scale space is used for this descriptor.

<sup>1</sup>The SMSLD implementation and evaluation code is available at: <https://github.com/bverhagen/SMSLD>



**SMSLD** The scale-invariant adaptation of the MSLD C++ implementation as described in Section 2.4. Scaled lines are generated using the parameters mentioned in Section 2.2.

**Btopo** The topological filter as described by Bay *et al.* [1]. While this filter is able to cope with both line segments and points, only line segments are used during the evaluation. For the topological filtering stage the default parameters as proposed by Bay *et al.* are used.

**KVLD** The K-VLD algorithm as implemented by Liu and Marlet [6]. K-VLD uses interest points, region descriptors and the K-connected graph matching algorithm.

**K** The implementation of the K-connected graph matching algorithm, as provided by Liu and Marlet [6]. The middle point, scale and orientation of the matched line segments are provided as input. Since the middle points of the line segments are not as accurately localized as interest points, a precision of 20 pixels is used for the graph matching. The VLD-consistency is raised to 1 and the contrast suppression is removed.

**ORSA** The original implementation of the *optimized random sampling algorithm* (ORSA) [10]. ORSA is used both for eliminating outliers and for estimating the homography between pairs of images. The middle points of the matched line segments are used as input points. Since the midpoints are not as accurately localized as interest points, a precision of 20 pixels is used.

**SIFT** The VLFeat 0.9.16 implementation [12] of SIFT. This method uses Lowe’s criterion (the ratio of the second closest to closest match) of 0.8 for pruning the matches.

In all the evaluations the fast approach for line segment scale extraction as described in Section 2.2 is used. During the evaluations the accuracy is defined as the ratio of the number of correct correspondences w.r.t. the total number of found correspondences. Since the midpoints of the line segments are not accurately localized and the corresponding lines (rather than only corresponding segments on the lines) are sufficient to accurately calculate the epipolar geometry, the correctness of a line segment correspondence is verified by its deviation from its actual corresponding line. Line segments with a difference in angle of less than 5 degrees and whose middle has an orthogonal distance of less than 5 pixels from the corresponding line are considered right matches.

### 3.2. Point matching benchmarks

First, the descriptors are tested using Mikolajczyk *et al.*’s dataset [9]. This dataset is often used for evaluating

and comparing point detectors and descriptors under image transformations like blurring, viewpoint changing, zoom, rotation, illumination and JPEG compression. The dataset contains 8 sequences of 6 images each testing one or more of these transformations. Within one sequence the image transformation is intensified for each consecutive image. Each image in the sequence is matched with the first image of that sequence. Note that the *Bark* sequence is not well suited for line segment matching. The *UBC* sequence on the other hand gives very good results. Both sequences are not shown in Figure 3. The K-VLD method is provided as a reference, it uses point and region descriptors in combination with the K-connected graph matching algorithm and achieves top performance on these benchmarks [6].

We compare Bay and SBay using the topological filter. MSLD and SMSLD are compared using no post-filter, the topological filter (Btopo) and the K-connected graph filtering method (K). The inlier rate is defined as the number of matches after K-filtering divided by the number of matches before K-filtering is applied.

As shown in Figure 3 in most of the setups adding scale-invariance to the Bay and MSLD line descriptors improves the performance by increasing the number of correct matches. While MSLD and SMSLD give a similar accuracy for the considered image transformations, post-filtering after SMSLD tends to give a higher accuracy increase than is the case with MSLD. This is a result of the redundancy present within the end result of the SMSLD algorithm. Note that Bay and SBay use color information and are vulnerable to illumination changes and repetitive color patterns. Moreover, Bay and SBay perform poorer than MSLD and SMSLD. Therefore, we do not report results based on the Bay descriptor with other filtering strategies.

### 3.3. Zoom sequence

The *Zoom* sequence tests specifically for the performance of the descriptor when confronted with scale changes. The scene depicts one zoomed in picture (picture 2) and 4 zoomed out pictures (3 - 4 - 5 - 6). The results are shown in Figure 4. The middle points of correct matches in image 1 are connected in green or yellow with their corresponding point on the support line in image 2, while wrong matches are connected in red with the middle point of their wrongly matched line segment. The correct extra correspondences found by SMSLD w.r.t. MSLD are marked in yellow.

As shown in Figure 4b, the original algorithms achieve a similar performance as the scale-invariant ones for small scale changes. However, even here the scale invariant methods already find more correct matches. For bigger scale changes the methods based on SMSLD have an accuracy that is about 10% to 15% higher than their scale-variant counterparts, while still being able to find more matches

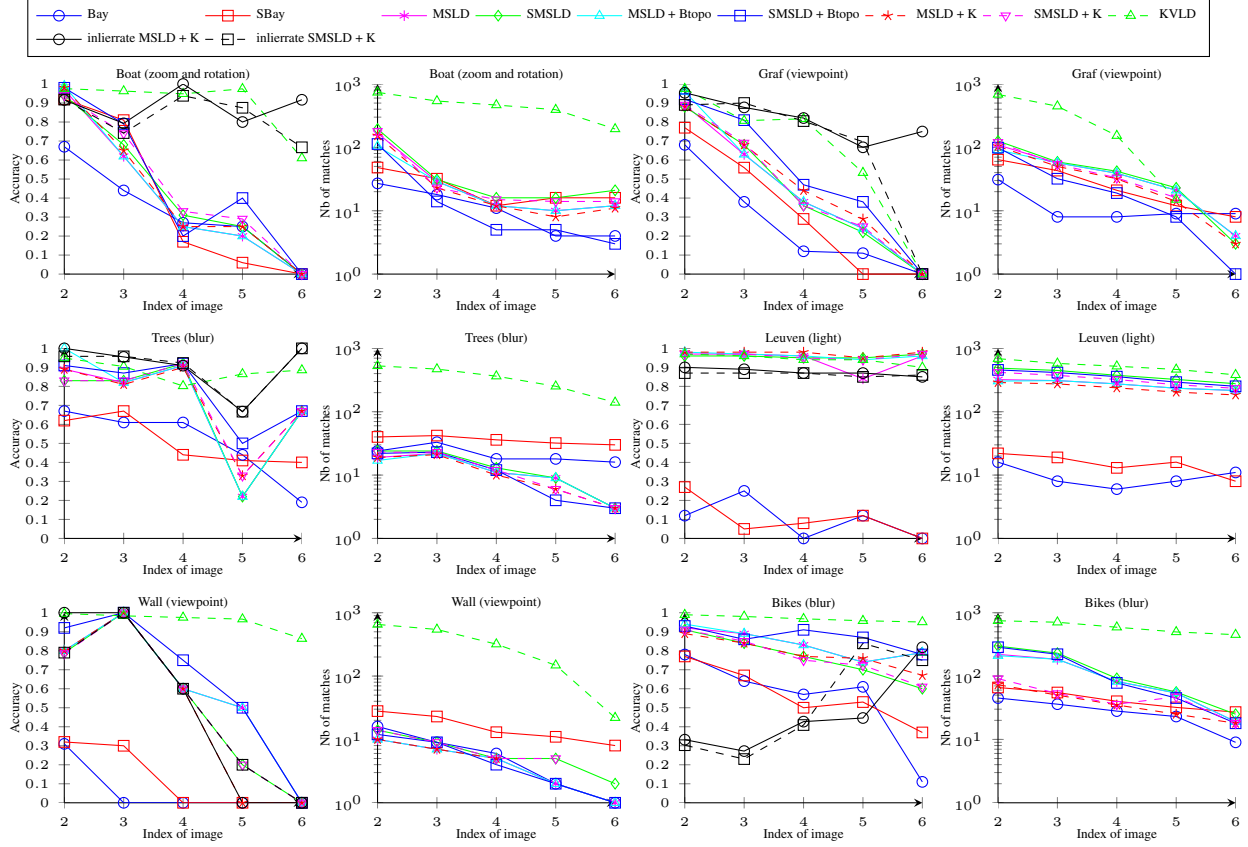


Figure 3: Performance evaluation on Mikolajczyk’s dataset. Note that the results for UBC and Bark are not included, very good results are obtained on UBC while Bark is not well suited for line segment matching. Note the logarithmic scale for the ‘number of matches’.

in all but the *SMSLD+Btopo* case. *SBay* finds more correspondences than *Bay*, but suffers from the repeatability of the textures within the image.

An important side note is that *SMSLD* is a trade-off between accuracy and execution time: *MSLD* combined with a scale space (*PMSLD*) of 5 levels and a scaling factor of 0.8 achieves on image pair 1 and 4 of the *Zoom* sequence an accuracy of 94% (89 found line segments) calculating 1145 (*MSLD*) descriptors, while *SMSLD* calculates 1004 (*SMSLD*) descriptors and has an accuracy of 83%. *SMSLD* combined with the same scale space (*PMSLD*) also achieves 94% (102 found line segments), calculating 1833 (*SMSLD*) descriptors.

### 3.4. Repeatability

The first two images of the *Castle-K19* sequence [11]<sup>2</sup> are used for testing the repeatability performance of the descriptors. We combine the SIFT, K-VLD, *MSLD* and *SMSLD* descriptors with *ORSA* to determine the dominant homography. The resulting inliers (green) and outliers

<sup>2</sup>The authors provide [11] as reference for using the *Castle-K19* sequence.

(blue) as considered by *ORSA* are provided in Figure 5. Since the size of the images of the *Castle-K19* sequence are 3072 px × 2048 px, the orthogonal distance threshold for scaled lines is set to 100 pixels and the minimum line segment length to 60 pixels. Noteworthy is that *ORSA* fails in estimating a good homography when used with SIFT, since the inliers are clustered on a tree. K-VLD is also provided as a reference. *SMSLD* outperforms *MSLD* in terms of (inlier) matches.

### 3.5. Difficult scenes

Line segments are especially useful in texture-less scenes. Figures 6 and 7 show the found correspondences for two texture-less indoor scenes. Figure 6 shows an example where line segment matching methods and point correspondence matching methods complement each other rather well: K-VLD finds most of its correspondences within the boxes on the wall, while line segment methods find correspondences in the surroundings. Note that where K-VLD fails (see Figure 7) in finding good correspondences, the line descriptors are able to find a few. In both images correct and incorrect correspondences are connected with blue

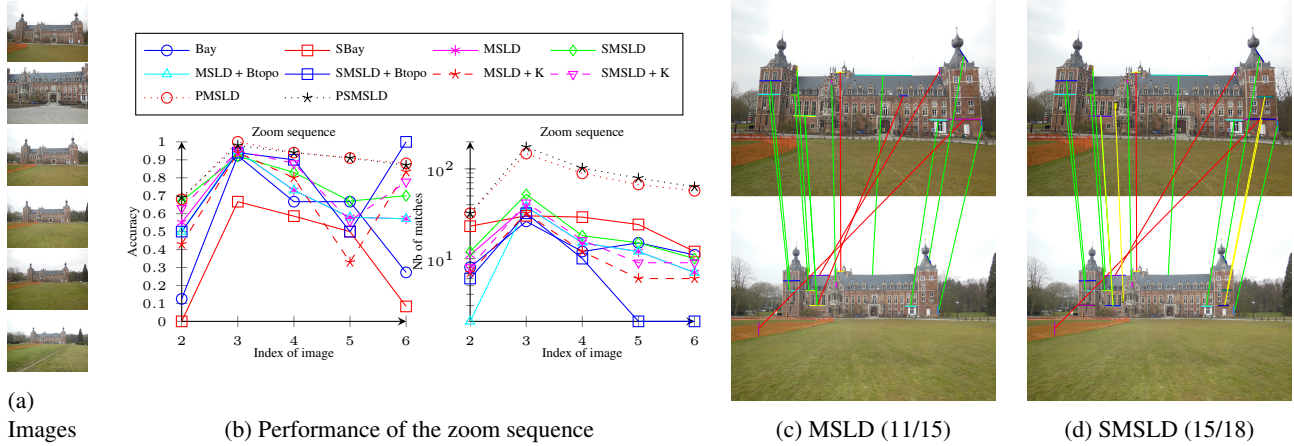


Figure 4: Results of the zoom sequence. (a) shows the 6 images of the sequence. (b) shows the accuracy and number of matches for each image in the sequence when matched with image 1. (c) and (d) show the result of MSLD and SMSLD respectively when matching images 1 and 4 of the sequence. For these we also report the number of correct matches/total number of matches between brackets. We depict with green lines the correct matches, with red the incorrect ones, and with yellow the extra correct correspondences found by SMSLD w.r.t. MSLD.

lines by their middle points. Yellow depicts the correspondences that are found only by one of the MSLD and SMSLD techniques (the bulk of correspondences in blue is common for both).

## 4. Conclusions

This paper proposes a method for adding scale-invariance to line segment descriptors. This method is illustrated on the scale-variant line segment descriptor proposed by Bay *et al.* [1] and the MSLD [14] descriptor resulting in the SBay and SMSLD algorithm respectively. The proposed descriptors are evaluated using standard and more challenging image datasets. The envisaged application of these descriptors is efficient wide baseline matching. For textured scenes, point correspondence methods are superior to line segment correspondence methods. However, for more difficult scenes the line segment matching methods are often a valuable addition to the point based methods. Scale-invariant methods offer the advantage of finding more correct matches resulting in better post-filtering or more accurate epipolar geometry estimations in the general case, while yielding a higher accuracy when confronted with bigger scale changes.

**Acknowledgments.** This work was partly supported by the EC FP7 Strep project ROVINA.

## References

- [1] H. Bay, V. Ferrari, and L. V. Gool. Wide-baseline stereo matching with line segments. In *CVPR*, 2005. 1, 3, 4, 5, 7
- [2] E. Bengoetxea. *Inexact Graph Matching Using Estimation of Distribution Algorithms*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, France, 2002. 1
- [3] V. Ferrari, T. Tuytelaars, and L. V. Gool. Simultaneous object recognition and segmentation by image exploration. In *ECCV*, 2004. 1
- [4] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 1
- [5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second edition, 2004. 1
- [6] Z. Liu and R. Marlet. Virtual line descriptor and semi-local matching method for reliable feature correspondence. In *BMVC*, 2012. 1, 4, 5
- [7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 2, 4
- [8] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, 2002. 1
- [9] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1-2):43–72, 2005. 5
- [10] L. Moisan, P. Moulon, and P. Monasse. Automatic homographic registration of a pair of images, with a contrario elimination of outliers. *Image Process. On Line*, 2012. 5
- [11] C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008. 6
- [12] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008. 5
- [13] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE TPAMI*, 32(4):722–732, 2010. 2, 4
- [14] Z. Wang, F. Wu, and Z. Hu. Mslr: A robust descriptor for line matching. *PR*, 42(5):941–953, 2009. 1, 2, 3, 4, 7
- [15] Z.-H. Wang, S.-S. Zhi, and H.-M. Liu. Mshs: The mean-standard deviation curve matching algorithm in hsv space. In *Machine Learning and Cybernetics, International Conference on*, volume 3, pages 1064–1069, 2012. 2
- [16] L. Zhang and R. Koch. Line matching using appearance similarities and geometric constraints. In *DAGM/OAGM Symposium*, pages 236–245, 2012. 2



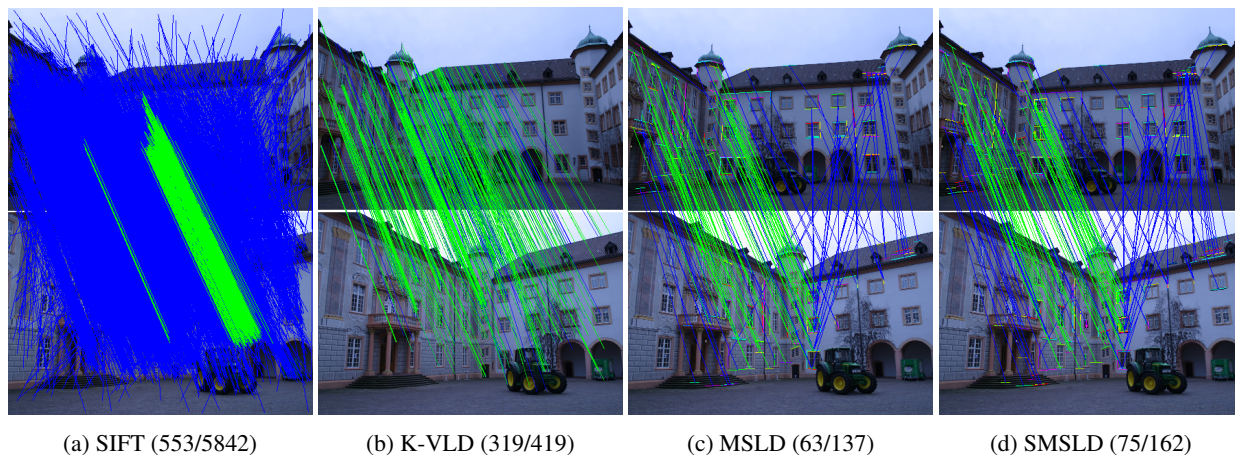


Figure 5: Result of the Castle-K19 sequence. ORSA is used for determining the homography. SIFT + ORSA is failing.

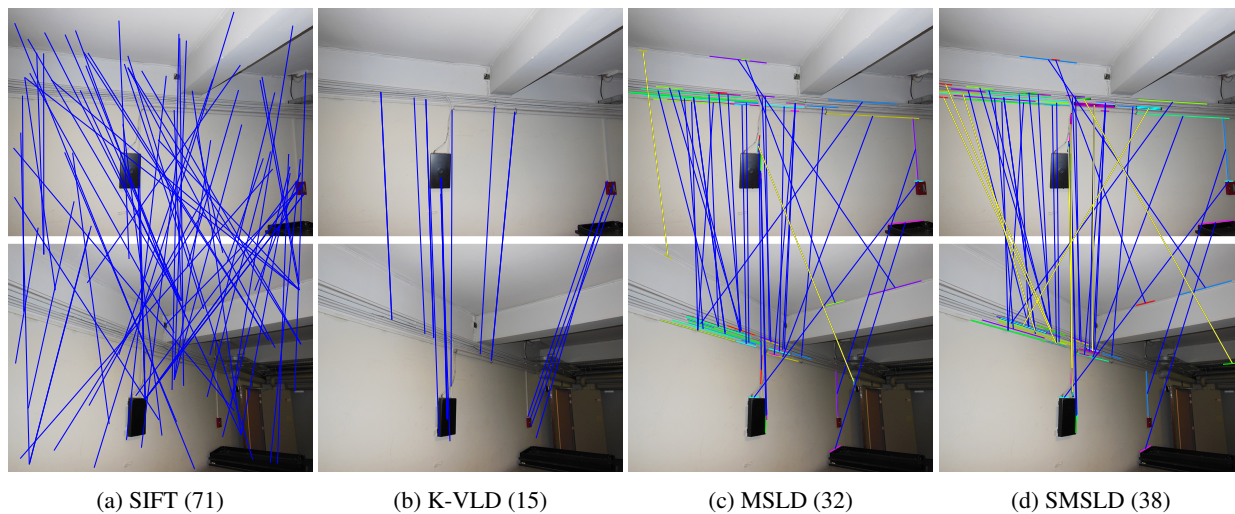


Figure 6: Difficult scene 1. The line segment descriptors provide a high number of correct matches.

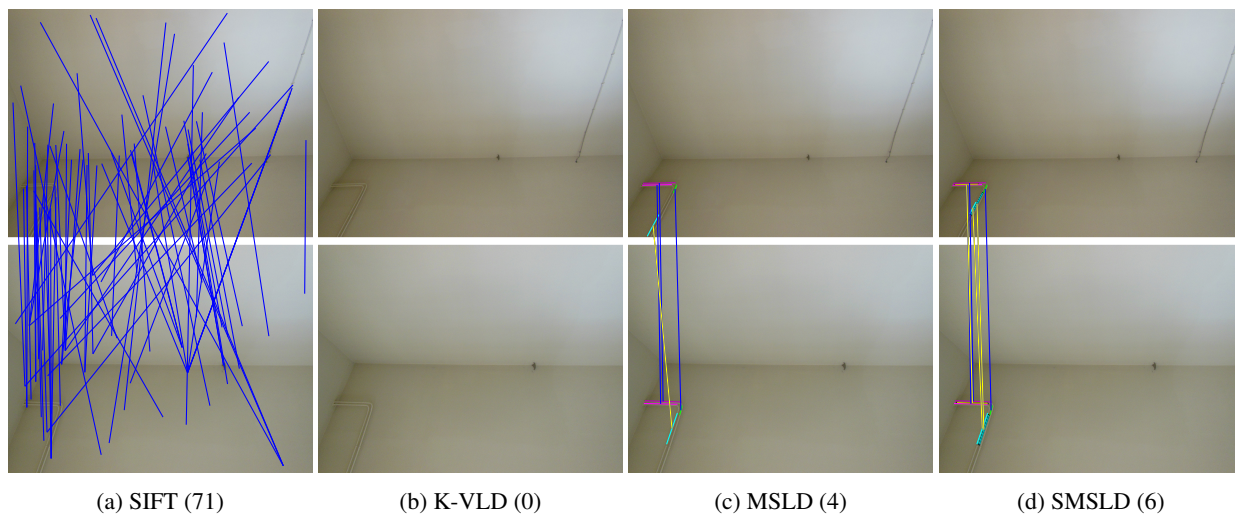


Figure 7: Difficult scene 2. K-VLD is failing. MSLD and SMSLD differ in the matches depicted with yellow.